R Notebook

```r
library(datasets)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(PCovR)
```

```
## Loading required package: GPArotation
## Loading required package: ThreeWay
## Loading required package: MASS
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
##
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

We first start by filtering the data and removing non-numerical columns.

```r
data <- read.csv("data_alexithymia2.csv", sep = ";")
str(data)
```

```
## 'data.frame':    122 obs. of  24 variables:
##  $ ID                 : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Sex                : chr  "F" "M" "M" "M" ...
##  $ Age                : int  19 19 18 18 18 23 21 19 23 18 ...
##  $ X.confused         : int  0 3 3 2 1 2 4 2 2 1 ...
##  $ X.right.words      : int  0 3 1 0 2 4 3 2 1 0 ...
##  $ X.sensations       : int  0 0 0 0 0 0 3 4 0 3 ...
##  $ X.describe         : int  3 3 3 4 4 0 2 2 3 4 ...
##  $ X.analyze.problems : int  1 4 3 4 4 3 0 2 2 3 ...
##  $ X.upset            : int  0 2 0 0 1 3 2 2 2 0 ...
```

```
##  $ X.puzzled          : int  0 1 0 2 0 2 2 3 2 0 ...
##  $ X.let.happen       : int  2 1 0 0 1 0 1 1 2 1 ...
##  $ X.identify         : int  0 2 1 0 1 3 1 3 1 0 ...
##  $ X.essential        : int  2 3 3 4 4 3 3 2 3 4 ...
##  $ X.feel.about.people: int  1 2 0 0 1 4 4 3 1 0 ...
##  $ X.describe.more    : int  0 0 0 0 2 1 0 2 2 0 ...
##  $ X.going.on         : int  0 1 0 0 0 3 3 4 1 0 ...
##  $ X.why.angry        : int  0 1 3 0 0 2 0 1 1 0 ...
##  $ X.daily.activities : int  2 1 1 4 0 2 1 1 1 0 ...
##  $ X.entertainment    : int  3 1 0 0 0 3 3 2 3 0 ...
##  $ X.reveal.feelings  : int  0 1 0 4 1 4 4 3 3 0 ...
##  $ X.close            : int  4 2 3 4 4 4 2 3 3 4 ...
##  $ X.useful           : int  2 3 2 4 3 3 4 3 3 4 ...
##  $ X.hidden.meanings  : int  2 1 0 4 4 3 1 2 0 1 ...
##  $ CESD               : int  0 23 46 11 8 18 26 16 13 3 ...
```

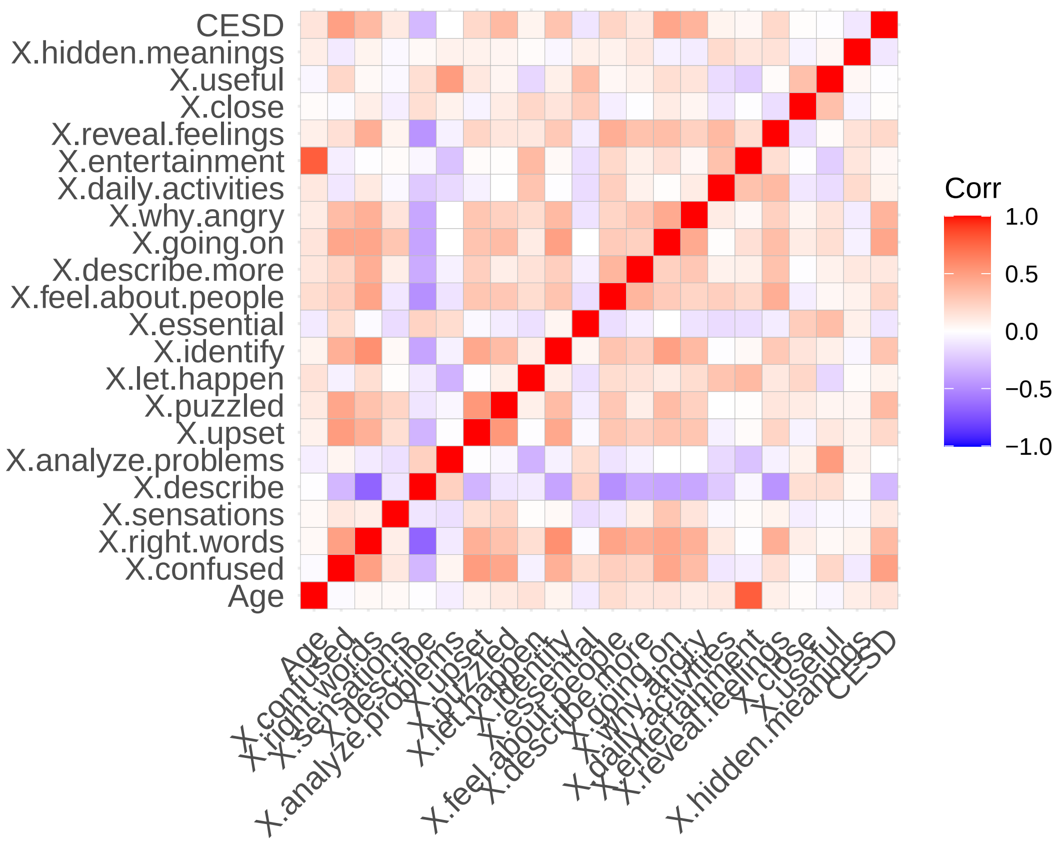
```r
numerical_data <- data[3:24] #remove non-numerical columns

pca.train <- data[1:nrow(numerical_data),]
pca.test <- data[-(1:nrow(numerical_data)),]
```

```r
data_normalized <- scale(numerical_data)
head(data_normalized)
```

```
##             Age X.confused X.right.words X.sensations X.describe
## [1,] -0.2969077 -1.5441476    -1.4037160   -0.5576343  0.6401771
## [2,] -0.2969077  1.0016093     0.9422203   -0.5576343  0.6401771
## [3,] -0.8811454  1.0016093    -0.6217372   -0.5576343  0.6401771
## [4,] -0.8811454  0.1530236    -1.4037160   -0.5576343  1.4622992
## [5,] -0.8811454 -0.6955620     0.1602416   -0.5576343  1.4622992
## [6,]  2.0400432  0.1530236     1.7241991   -0.5576343 -1.8261893
##      X.analyze.problems    X.upset  X.puzzled X.let.happen X.identify
## [1,]         -1.3996685 -1.2399900 -1.0014023    0.6463805 -1.2876255
## [2,]          1.3844547  0.3036710 -0.1224737   -0.2298242  0.2832776
## [3,]          0.4564136 -1.2399900 -1.0014023   -1.1060288 -0.5021739
## [4,]          1.3844547 -1.2399900  0.7564550   -1.1060288 -1.2876255
## [5,]          1.3844547 -0.4681595 -1.0014023   -0.2298242 -0.5021739
## [6,]          0.4564136  1.0755015  0.7564550   -1.1060288  1.0687292
##      X.essential X.feel.about.people X.describe.more  X.going.on X.why.angry
## [1,]  -0.6427154          -0.5938360    -0.863389537 -0.81104341  -1.0250323
## [2,]   0.2585637           0.2294366    -0.863389537  0.01351739  -0.2076863
## [3,]   0.2585637          -1.4171086    -0.863389537 -0.81104341   1.4270057
## [4,]   1.1598427          -1.4171086    -0.863389537 -0.81104341  -1.0250323
## [5,]   1.1598427          -0.5938360     0.849350683 -0.81104341  -1.0250323
## [6,]   0.2585637           1.8759818    -0.007019427  1.66263899   0.6096597
##      X.daily.activities X.entertainment X.reveal.feelings    X.close   X.useful
## [1,]          0.2731044       1.0298571        -1.1374838  1.0378797 -0.4277312
## [2,]         -0.5395477      -0.5505525        -0.4470708 -0.7581674  0.4877637
## [3,]         -0.5395477      -1.3407574        -1.1374838  0.1398561 -0.4277312
## [4,]          1.8984087      -1.3407574         1.6241685  1.0378797  1.4032586
## [5,]         -1.3521999      -1.3407574        -0.4470708  1.0378797  0.4877637
## [6,]          0.2731044       1.0298571         1.6241685  1.0378797  0.4877637
##      X.hidden.meanings       CESD
## [1,]         0.7697136 -1.5136856
## [2,]        -0.2084641  0.6005015
```

```
## [3,]        -1.1866419  2.7146886
## [4,]         2.7260692 -0.5025527
## [5,]         2.7260692 -0.7783162
## [6,]         1.7478914  0.1408956
```

```
library(ggcorrplot)
corr_matrix <- cor(data_normalized)
ggcorrplot(corr_matrix)
```



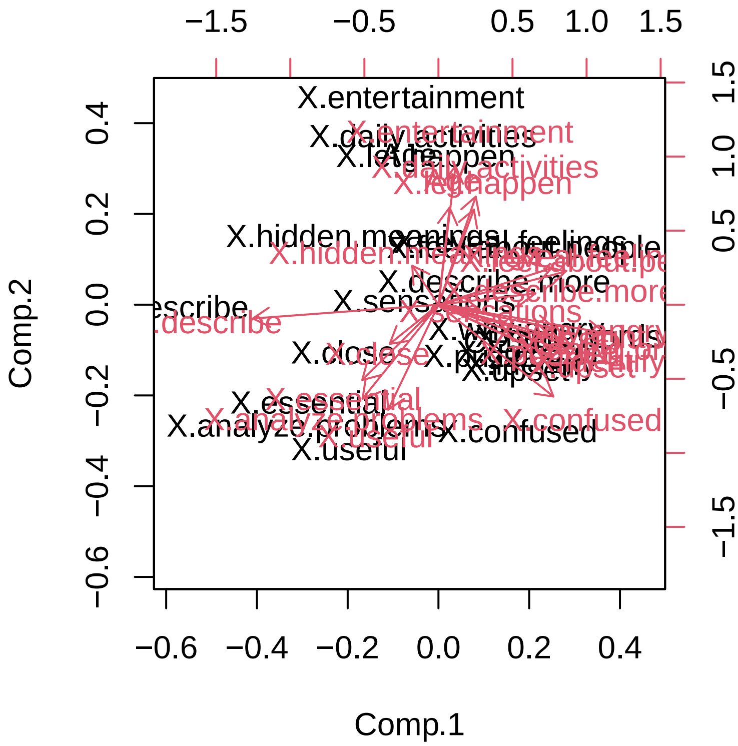

```
pca_results<- princomp(corr_matrix)
summary(pca_results)
```

```
## Importance of components:
##                           Comp.1    Comp.2     Comp.3     Comp.4     Comp.5
## Standard deviation     0.8325561 0.5674647 0.33543793 0.28585191 0.25897969
## Proportion of Variance 0.4451410 0.2067989 0.07225948 0.05247503 0.04307268
## Cumulative Proportion  0.4451410 0.6519398 0.72419930 0.77667433 0.81974701
##                            Comp.6     Comp.7     Comp.8     Comp.9    Comp.10
## Standard deviation     0.23778461 0.20845694 0.18851774 0.16365510 0.15554210
## Proportion of Variance 0.03631098 0.02790636 0.02282312 0.01720005 0.01553697
## Cumulative Proportion  0.85605799 0.88396436 0.90678747 0.92398752 0.93952449
##                           Comp.11    Comp.12     Comp.13    Comp.14     Comp.15
## Standard deviation     0.13931579 0.13070954 0.109893096 0.10157018 0.095974031
## Proportion of Variance 0.01246439 0.01097198 0.007755527 0.00662526 0.005915316
## Cumulative Proportion  0.95198889 0.96296087 0.970716396 0.97734166 0.983256973
##                            Comp.16     Comp.17     Comp.18     Comp.19
## Standard deviation     0.089363618 0.079196915 0.069431688 0.064333557
## Proportion of Variance 0.005128519 0.004027977 0.003095893 0.002657943
```

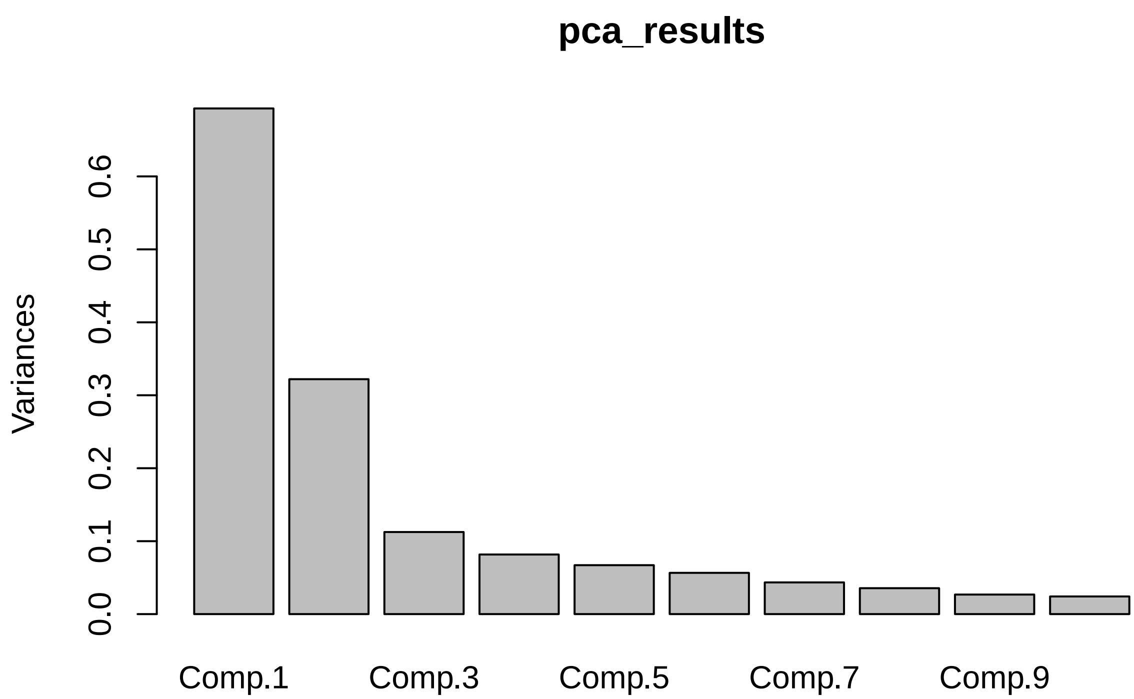
```
## Cumulative Proportion  0.988385492 0.992413470 0.995509363 0.998167305
##                            Comp.20      Comp.21      Comp.22
## Standard deviation     0.047655997 0.0241387681 2.639495e-09
## Proportion of Variance 0.001458497 0.0003741974 4.474167e-18
## Cumulative Proportion  0.999625803 1.0000000000 1.000000e+00
```

Principle Component Analysis (PCA) is an important technique used in dimensionality reduction. It is particularly relevant for datasets where the number of dependent variables are really high. It does this by shrinking the variables down to a number of components that are more significant. The PCA results can help indicate which components are more relevant than others. A principle component analysis (PCA) of our dataset is done above. This shows that the first component explains 44.5% of the total variance. This means that almost half of our dataset can be represented by the first component. The second component explains 65% of our variance and so on.

```r
biplot(pca_results)
```




```r
plot(pca_results)
```



A biplot and bargraph both show that the first two components are most significant. A principle component analysis (PCA) of our dataset is done above. This shows that the first component explains more than 60% of the total variance. The second component a little over 30% of our variance and so on. Thus we pick the first two components.

A closer look is given below.

```
#Rotation
pca_result <- prcomp(numerical_data[, -5], scale. = TRUE)  # Exclude the species column (5th column)

# Extract loadings
loadings <- pca_result$rotation

# Print loadings for PC1 and PC2
print(loadings[, 1:2])
```

```
##                            PC1          PC2
## Age                 0.11112494 -0.304848075
## X.confused          0.30678302  0.248104861
## X.right.words       0.35672657  0.040793978
## X.sensations        0.10478879 -0.003014913
## X.analyze.problems -0.05093660  0.333132709
## X.upset             0.29355650  0.121607533
## X.puzzled           0.27408773  0.090862319
## X.let.happen        0.11215194 -0.309275596
## X.identify          0.31876109  0.105567840
## X.essential        -0.04431382  0.275899603
## X.feel.about.people 0.27592101 -0.140074223
## X.describe.more     0.23303742 -0.060959169
## X.going.on          0.33067260  0.079092476
## X.why.angry         0.28443736  0.036485567
## X.daily.activities  0.08129515 -0.342250036
## X.entertainment     0.09697418 -0.438065749
## X.reveal.feelings   0.24774936 -0.143350294
```

```
## X.close              0.02356318  0.149658909
## X.useful             0.05202018  0.361872792
## X.hidden.meanings    0.01747772 -0.100522255
## CESD                 0.27131017  0.054185173
```

The above table can be interpreted as follows. Higher positive loadings such as the one of X.confused wiht PC1 (0.306) show a strong positive relationship between X.confused and PC1. On the other hand a high negative loading show a strong negative relationship, for example X.daily.activities with PC2 (-0.3).

```r
#install.packages("psych")
library(psych)
```

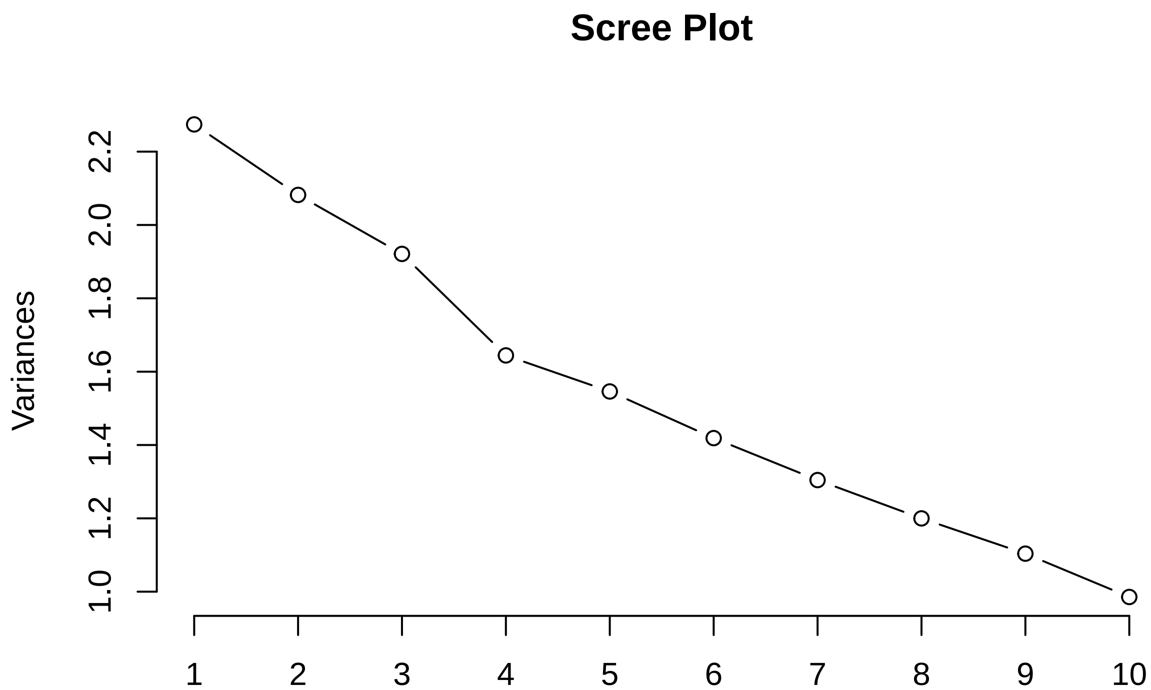
```
##
## Attaching package: 'psych'

## The following objects are masked from 'package:ThreeWay':
##
##     phi, tr

## The following objects are masked from 'package:GPArotation':
##
##     equamax, varimin

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```r
library(GPArotation)
# Load your data
#data <- read.csv("data_alexithymia2.csv", sep = ";")

# Rotate variables using the varimax rotation

data <- as.matrix(numerical_data)
rotated_data <- varimax(data)


# Perform PCA on rotated data
pca_result <- prcomp(rotated_data$loadings, scale. = TRUE)

# Print summary of PCA
summary(pca_result)
```

```
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.5081 1.44291 1.38605 1.28230 1.24336 1.19115 1.14209
## Proportion of Variance 0.1034 0.09464 0.08732 0.07474 0.07027 0.06449 0.05929
## Cumulative Proportion  0.1034 0.19801 0.28533 0.36007 0.43035 0.49484 0.55413
##                            PC8     PC9   PC10    PC11    PC12   PC13   PC14
## Standard deviation     1.09539 1.05057 0.9928 0.94905 0.90776 0.8899 0.7945
## Proportion of Variance 0.05454 0.05017 0.0448 0.04094 0.03746 0.0360 0.0287
## Cumulative Proportion  0.60867 0.65884 0.7036 0.74458 0.78203 0.8180 0.8467
##                           PC15    PC16    PC17    PC18    PC19    PC20   PC21
## Standard deviation     0.77020 0.76154 0.72466 0.71655 0.64402 0.58786 0.5648
## Proportion of Variance 0.02696 0.02636 0.02387 0.02334 0.01885 0.01571 0.0145
## Cumulative Proportion  0.87369 0.90005 0.92392 0.94726 0.96611 0.98182 0.9963
##                           PC22
```

```
## Standard deviation     0.28463
## Proportion of Variance 0.00368
## Cumulative Proportion  1.00000
```

```r
plot(pca_result, type = "l", main = "Scree Plot")
```




```r
# Kaiser's criterion
eigenvalues <- pca_result$sdev^2
selected_components <- sum(eigenvalues > 1)

# Cumulative proportion of variance
cumulative_variance <- cumsum(pca_result$sdev^2 / sum(pca_result$sdev^2))
selected_components <- sum(cumulative_variance <= 0.8)  # Adjust the threshold as needed
```

A rotation and scree plot give us a clearer understanding of which compoennts to pick. Since PC1 and PC2 contribute more to the overall variance, we pick them.